



datainnovations.com

Getting Started with Rules - Rules 101

Getting Started with Instrument Manager Rules

Agenda

1 The Basics

2 Creating Rules

3 Writing Rules

Agenda

1 The Basics

2 Creating Rules

3 Writing Rules

The Basics

Getting Started with Rules

- Rules are configuration-based
 - Allows for rules to be shared across multiple analyzers (one to many)
- Rules can be written against the information in IM
 - Review the Specimen Event Log (SEL) for available data elements
 - Example: Location-based rules

Location - Bed
Location - Facility
Location - Room
Location - Ward
Location Nurse Station

```
Record was a request message
Message Data:
Group Data:
Group Identifier = '105,701'
Group Entry Type = 'C'
Group Sequence Number = '6789'

Patient Data:
Patient ID = 'XY106789'
Patient Name = 'Lab Intel,Data'
Sex = 'M'
Date of Birth = '5/21/1962'
Location - Facility = 'First Floor'
Race = 'Cauc'
```

The Basics Continued

- Order of Operation

- IM does not adhere to PEMDAS (Parenthesis, Exponents, Multiplication, Division, Addition Subtraction)
 - Will perform operations in the order that they appear
 - $3+5*3-2 = 22$ in IM rather than 16 if you follow PEMDAS
 - IM uses parentheses to force order of operation
 - $3+(5*3)-2 = 16$ in IM
 - Parentheses also used to group “thoughts” in IM together to make sure statements are properly evaluated

- Organization of Rules

- Rules fire from top to bottom, left to right
- Rules should be organized in a logical fashion
 - Example: Having a rule fire to perform a calculation on an invalid result

The Basics Continued

- Overview of the Rules Screen

- Test/In Validation versus Live

- Red versus Black

- If/Then/Else

- Initially, lots of “demand” for Else

- Very rare to actually use Else...very few “it’s this or it’s that” in the Laboratory

- Request or Result Rule?

- Location of Rules

- Lots of places to write rules

- The vast majority of rules still written in Incoming Result>Before Message Queued Internally

The Basics Continued

- Three basic levels of data elements in Rules
 - Patient, Specimen and Test
 - General Data Elements as well that do not fit in to those categories
 - Patient and Specimen data elements are usually persistent
 - Test Data Elements can be persistent but some/most are not

Agenda

1 The Basics

2 Creating Rules

3 Writing Rules

Creating Rules

- Three ways to write rules
 - Drag and Drop
 - Double-Click
 - Free Text
- Likely to use a combination of those methods
 - Free text is the fastest

Creating Rules Continued

- Some users may find “sketching” out their rules helps
 - Write your rules out on paper first
- Parentheses are used primarily in the IF statement
 - Used in the THEN statement **only** if doing a calculation to force order of operation
 - Also used to encase “conditions” that need to be true
- Add versus Set
 - “Add” adds on to the field you specify and leaves the original contents in place. “Set” replaces what was there with what you tell it to set
- Add Test versus Order Test
 - Add Test adds that test on to your existing message
 - Used primarily in Result messages to add on a test that was not part of the original result message, such as a calculated value

Creating Rules Continued

- Add Test versus Order Test Continued
 - Order Test used to order a new test that was not part of the original message
 - Most commonly used in Request rules, or to order a reflex test

Agenda

1 The Basics

2 Creating Rules

3 Writing Rules

Writing Rules

- Using the Drag and Drop Method
 - Write a rule that adds on test “HIL” if a Glucose is ordered
 - First, determine if this a Request or Result rule?
 - Where are we going to write this rule?
 - User Input Value
 - Allows you to define something specific, such as a test code, error code, test result
 - If: ({Test Ordered} “GLU”)
Then: {Add Test} “HIL”

Writing Rules Continued

- Using the Double-Click Method

- Write a rule that adds a test comment of “Severe Hemolysis Present – Result may be elevated” to Potassium if Hemolysis is greater than or equal to 3+
- What type of rule is this, request or result?
- Where do you write it?
- What 2 things do I need to check for this rule?
 - Is my hemolysis result \geq 3+?
 - Do I have a K result present?
- Do I care what my Potassium result is?
- If: ({Test Resulted} “K”) {AND} ({Hemolysis} \geq “3+”)
Then: {Add} {Test Comment(s)} “Severe Hemolysis Present – Result may be elevated”
{On Test} “K”

Writing Rules Continued

- Using the Free Text Method

- If Sodium, Chloride and Bicarbonate results are present, add test AGAP and use the formula = Na – (Cl + HCO3)
- What type of rule? Where do you write it?
- What do you need to check?
 - Since we are doing a calculation, we need to check that all values are numeric
 - We do not need to check to make sure all results are present and doing the numeric check takes care of that as well
- IM is not very smart when it comes to multiple items...you will need to “explain” to it what you are trying to do very literally.
 - You can’t just say “If Na and Cl and HCO3 are numeric”.
 - You’ll need to say “If Na is numeric AND Cl is numeric and HCO3 is numeric”

Writing Rules Continued

- Using the Free Text Method Continued

- If: (({Result} {On Test} "NA" {Is Numeric}) {AND} ({Result} {On Test} "CL" {Is Numeric}) {AND} ({Result} {On Test} "HCO3" {Is Numeric}))
Then: {Add Test} "AGAP" {AND} {Set} {Result} {On Test} "AGAP" = {Result} {On Test} "NA" - ({Result} {On Test} "CL" + {Result} {On Test} "HCO3")

Writing Rules Continued

- Saving Rules

- Save early, save often!
- Save to Test/In Validation first, then move to Live
- Saving rules checks for syntactical errors in the rules, not for “bad” rules
- Warnings Versus Errors
 - Errors cannot be moved to Live
 - Errors can usually be deciphered, but occasionally can be quite cryptic in nature
 - Warnings are just that...something needs your attention but may be okay
- Typical Issues
 - Parentheses
 - Is Numeric
 - Value Lists incomplete or not defined



datainnovations.com

Testing and Validation

Agenda

1 Testing Strategy

2 The Test Engine

3 Testing Your Rules

Agenda

1 Testing Strategy

2 The Test Engine

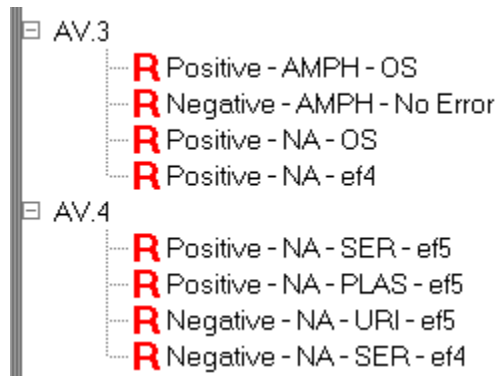
3 Testing Your Rules

Testing Strategy

- Overall Testing Strategy for IM Rules
 - Test what is needed
 - Look at the data elements the rule is using
 - Test multiple scenarios
 - Test in the affirmative, negative and the absurd
 - Testing of value lists
 - Test items in the beginning, the middle and the end of the value list
 - Testing the logic of the rule, not the value list
 - Do not test rules in a vacuum
 - Rules can be tested individually, but should also be tested as part of the whole rule set

Testing Strategy

- Overall Testing Strategy for IM Rules Continued
 - Spend the time to build a good set of test scenarios and cases
 - Saves time in the long run
 - Build your test cases to match your rule schema



- Print/save your audit trails
 - Regulators love to see these

Agenda

1 Testing Strategy

2 The Test Engine

3 Testing Your Rules

The Test Engine

- The Test Engine
 - One of the most powerful tools in IM
 - Virtually eliminates the need to do “wet-testing” with the analyzer
 - Allows for the simulation of any result; allows for easily testing the “edges” of rules
 - Easily create and recreate testing scenarios for regulatory purposes
- Tips and Tricks for the Test Engine
 - Age
 - Items needed to calculate a patient age:
 - DOB
 - Collection Date and Time
 - How old the sample is, not the patient

The Test Engine

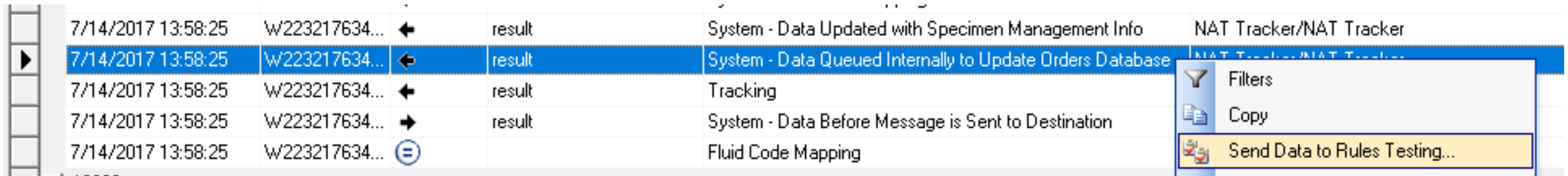
- Tips and Tricks for the Test Engine Continued
 - Adding or Removing Fields
 - Not all fields are displayed
 - Use the Field Chooser to add or remove unneeded fields
 - Copying/Duplicating Test Cases
 - Test cases can be copied/pasted using standard Windows commands
 - Test cases can also be copied to other configurations as needed
 - Test cases can be imported and exported as well
 - Testing Scenarios
 - Allows for multiple test cases to be run at once
 - Test cases can build on each other; Something “set” in the first case can be used by the second test case
 - For this to happen, the test cases need to match up
 - Usually requires Patient ID and possibly the request ID being set in the test cases

The Test Engine

- Tips and Tricks for the Test Engine Continued

- Send Data to Rules Testing

- Allows you to send real-world examples to the testing engine
 - Realistic examples from the analyzer eliminate any testing bias that might exist
 - From the SEL, choose the appropriate entry to send to Rules Testing
 - Most likely you will want to choose the “System - Data Queued Internally to Update Orders Database”
 - This is the entry that has all of the data possible; includes all of the information from the analyzer as well as the information that IM “knows” for the sample
 - Cannot have the Rules configuration screen open to the configuration you are trying to copy to



The screenshot shows a table with five rows. The second row is selected and highlighted in blue. A context menu is open over this row, showing options: Filters, Copy, and Send Data to Rules Testing... (highlighted in yellow).

7/14/2017 13:58:25	W223217634...	←	result	System - Data Updated with Specimen Management Info	NAT Tracker/NAT Tracker
7/14/2017 13:58:25	W223217634...	←	result	System - Data Queued Internally to Update Orders Database	NAT Tracker/NAT Tracker
7/14/2017 13:58:25	W223217634...	←	result	Tracking	
7/14/2017 13:58:25	W223217634...	→	result	System - Data Before Message is Sent to Destination	
7/14/2017 13:58:25	W223217634...	⊕		Fluid Code Mapping	

The Test Engine

- Tips and Tricks for the Test Engine Continued
 - Send Data to Rules Testing Continued

The screenshot displays a software interface with three main panels. The left panel, titled 'Test Scenarios', contains a toolbar with various icons and a list of scenarios. One scenario is expanded, showing 'Imported from Specimen Event Log' with a sub-entry 'From Specimen ID: W223217634590'. The right panel is divided into three sections: 'Patient Information', 'Specimen Information', and 'Test Information', each containing a table of data.

Patient ID	Sex	Date of Birth	Location - Facility

Specimen ID	Request ID	Last Run ID	Last Run Date/Time	Specimen
W22321763...	W2232176...	1	7/19/2017 03:55:24	205510

Test Code	Result	Result Date/Time	Test Instrument ID	Test Name
ULTRIO	Nonr...	7/13/2017 02:52:53	NAT Tracker	ULTRIO
*				

The Test Engine

- Tips and Tricks for the Test Engine Continued

- Send Data to Test Engine Continued

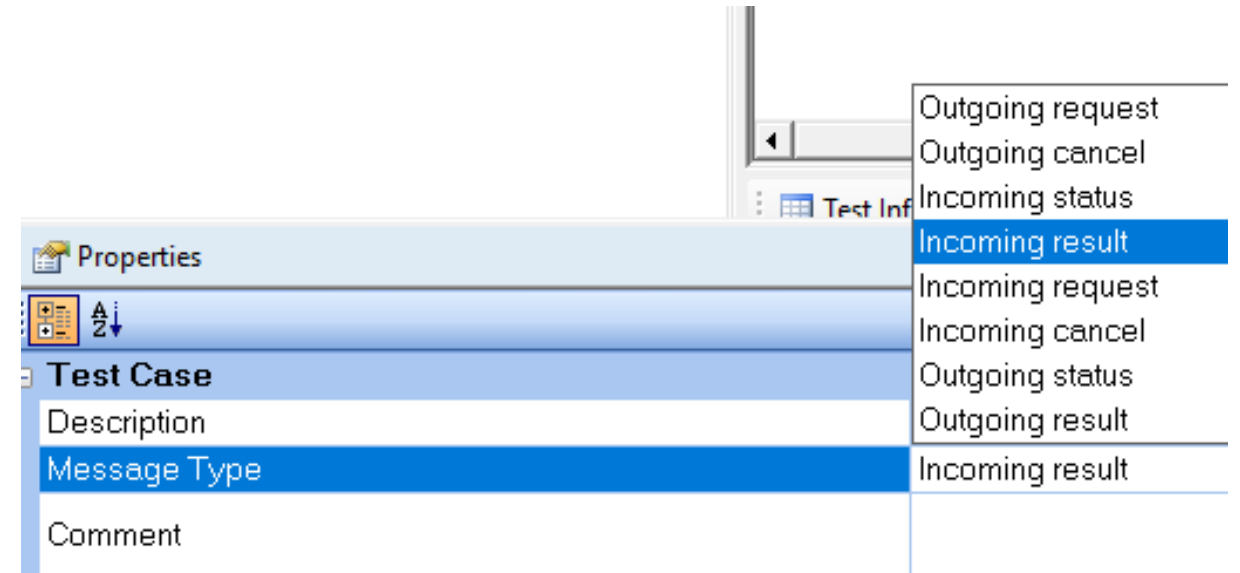
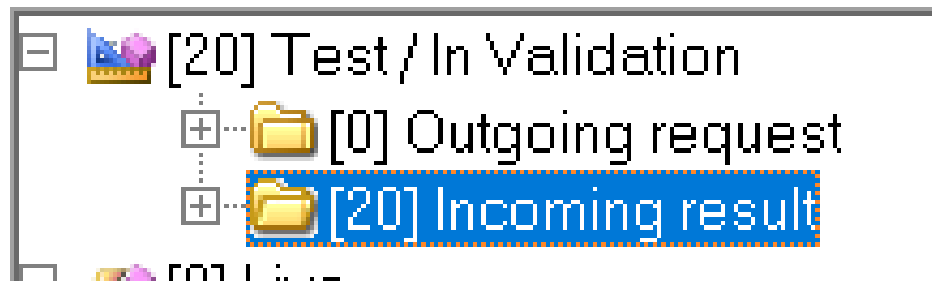
- You do not want to use Audit Trails for this purpose
 - Audit trails reflect the state of the message AFTER rules have already fired

- Executing Test Cases

- If you right-click, always click on the red letter of the test case
 - Right-clicking on the test case/scenario name usually causes IM to think you are trying to rename the test case
 - Depending on what you are trying to test, you may need to alter the properties of the test case
 - Always match the Message Type of the Test Case to the folder your rule is located in

The Test Engine

- Tips and Tricks for the Test Engine Continued
 - Executing Test Cases Continued



The Test Engine

- Tips and Tricks for the Test Engine Continued

- Testing complicated rules

- May require that you break down the rule in to smaller pieces first to test
 - Add back in additional pieces once you have the smaller piece(s) working
 - Cut/Copy/Paste

- Parentheses

- Should always have an even number of parentheses
 - For every open parentheses, you need a close parentheses
 - If you get a parentheses error, add/remove 1 at a time
 - Best method I have found for “counting” parentheses?



Agenda

1 Testing Strategy

2 The Test Engine

3 Testing Your Rules

Testing Your Rules

- Let's go back and test the rules we wrote earlier
- The first rule was this:
 - If: {Test Ordered} "GLU"
Then: {Add Test} "HIL"
 - What type of test case is this? Request or Result?
 - What data elements do we need in our test case?
 - DOB?
 - Collection Date/Time?
 - Patient ID?

Testing Your Rules

- The second rule was:
 - If: ({Test Resulted} “K”) {AND} ({Hemolysis >= “3+” })
Then: {Add} {Test Comment(s)} “Severe Hemolysis Present – Result may be elevated”
{On Test} “K”
 - What type of test case is this? Request or Result?
 - What data elements do we need in our test case?

Testing Your Rules

- The third rule was:

- If: (({Result} {On Test} "NA" {Is Numeric}) {AND} ({Result} {On Test} "CL" {Is Numeric}) {AND} ({Result} {On Test} "HCO3" {Is Numeric}))
Then: {Add Test} "AGAP" {AND} {Set} {Result} {On Test} "AGAP" = {Result} {On Test} "NA" - ({Result} {On Test} "CL" + {Result} {On Test} "HCO3")
- What type of test case is this? Request or Result?
- What data elements do we need in our test case?

Questions?

Thank you for your time!

2019 Rules Training Workshops –

Seats Available in Vermont...

- March 19 - 21
- April 23 - 25
- May 21 – 23
- June 25 - 27

- **Course objectives and syllabus**

<http://www.datainnovations.com/training>

- **To purchase seats contact DI sales:**

northamerica-sales@datainnovations.com

or by phone at 802 – 658 - 2850.